Overview

A trustless notary is a (set of) smart contract(s) that takes on the sybil resistance responsibilities that are normally delegated to elected notaries. Given the size and scale of the Filecoin network, it is unreasonable to imagine that any human-in-the-loop process will meaningfully scale as the network scales.

The design of this system explicitly aims to solve for some shortcomings in Fil+, while retaining its core strengths. This system can be further enhanced to reward different participants.

Limitations of existing system:

- "Useful" or "valuable" are inherently subjective terms, and its is bad for the network to encourage these sorts of decisions in a way that infringes on the rights of any individual node. (An individual node should be able to express their own values (eg complying with local regulations), but we should minimize/remove the ability for a third party to enforce a decision between two consenting parties.)
- Not all data that is useful is public, nor is all useful data meant to be accessible by all (healthcare data). Any system that relies on knowing what the data is, will inherently be limiting.

Strengths of existing system:

- Keeping storage costs low for clients

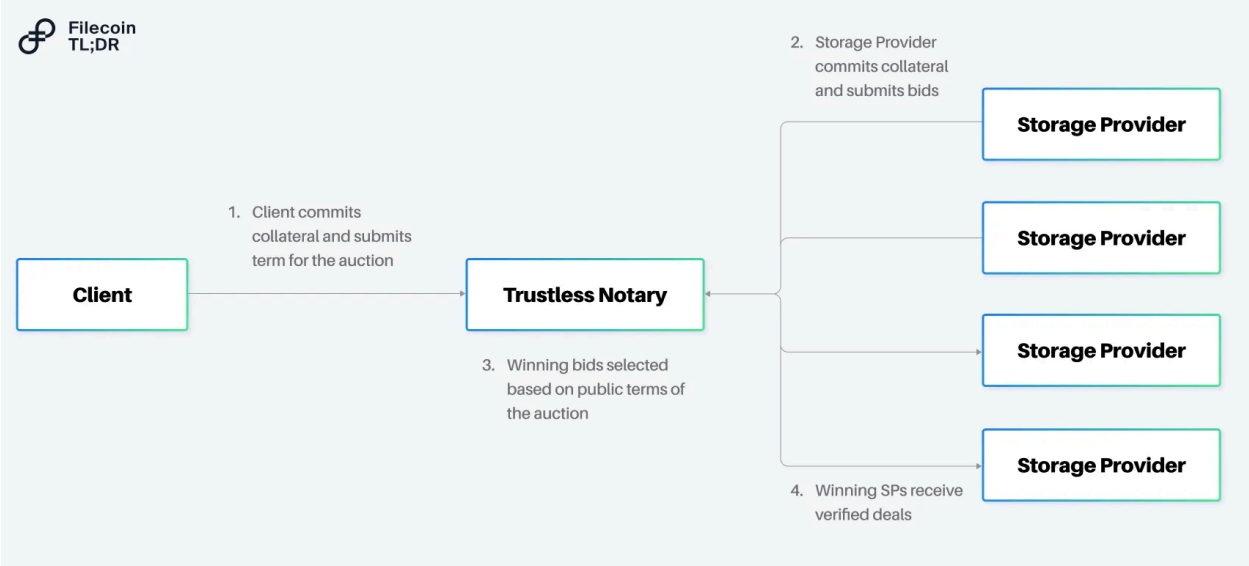
Target properties of a new system:

- The cost of storage on the network for clients should be minimized
- Subjective decisions about the legitimacy or desirability of datasets should be avoided at a datacap allocation level (nodes may make their own local decisions)
- Self dealing / storing data with oneself should be significantly less attractive than working with clients.
- DataCap allocation should be permissionless, without risking a sybil attack

Benefits for a permissionless design:

- Permissionless systems scale better than permissioned ones - at the volumes of data we'd like onboarded, this seems like a necessary step
- Social layers of trust have their own limits
- New types of data should be eligible for datacap without requiring compromising on privacy.
- If we solve FIL+'s issues, it should *further* reward SPs working with real clients, potentially help incentivize more OSS tooling sitting on top to simplify onboarding, and more

Rough Design



The motivation with this design is to lean into the fact that the primary role of the notary is to break up a sybil-y attack (a storage provider pretending to be a client to just store data with themselves). The notary was verifying *clients* not *data* - so instead this proposal focuses on designing a system where we can make it irrational (or substantially less attractive) for a malicious client to participate.

There are several types of participants in this new flow:

- Client
- (Optional) Onboarding agent
- Trustless Notary
- Storage Providers

The basic design is as follows:

- A client (or a delegated onboarding agent) can stake some amount of funds with the trustless notary to get access to datacap. A number of factors might impact the collateral requirements for a client:
    - The amount of collateral that the client must stake is proportional to their onboarding rate (ie how much datacap they'd like to consume in some period of time)
    - Restrictive requirements (hyperfiltering to a small set of SPs) for onboarding
    - Lack of history of a client with the auction (reputation might lower the cost of collateral for a client over time)
- The client can specify specific requirements of the types of storage providers that are eligible to bid to store their data
- The client can run an auction via the trustless notary to get their data stored
    - The client is required to ensure that their data is transferred (so if a deal is struck that they send the data through), and if they fail to do in a time frame they lose their collateral)

- On the other side, storage providers must also stake funds to participate in the auction.
- During the auction, eligible storage providers can bid at any price for the deal (including negative prices)
    - Storage providers who bid during a deal are committing to starting the deal in some window after the close of the auction, sacrificing their collateral if they fail to do so.
- Payments (either from client paying SP, or from SP paying client) flow to the trustless notary contract who can redistribute funds accordingly.
    - A trustless notary might have a tokenized version of an endowment that it might use to allocate across its different stakeholder groups to incentivize participation in whatever group is lacking (Clients, SPs, Onboarding Agents)

Benefits in this flow:

- Anyone can front the collateral to onboard a client - this could be the client themselves, an onboarding agent (who might make it their business), a DataDAO, or a group of storage providers.
- No humans in the loop - this flow should allow smart contracts to permanently stake capital with a trustless notary (again, this locked capital can be the staked representations of Filecoin meaning its not holding an opportunity cost) to get a fixed onboarding rate.
- If designed correctly, the incentives should make it possible (but less attractive) for storage providers to self deal. They still may choose to do so (perhaps a large SP winning BD for a large enterprise customer) - but they might now have a stronger incentive to work with a network of other providers to get paid an onboarding fee to recoup.
    - Equally in regions where there are few storage providers - there's now an incentive to grow operations if there is relatively less competition for more capital efficient deals.
- Tokenizing endowments and redistributing rewards (from some portion of negative bids, storage fees) offer some interesting opportunities:
    - Incentives to bring more specialized data onboarding agents / clients on to the network (and a business model for them)
    - Subsidies for SPs that lose deals but participate in the auction system (redistributing negative bids across others)

Some side notes here:

- The specifics of how collateral should be calculated, duration its locked, etc etc - need to be sorted out.
- The collateral itself can be staked representations of Filecoin - and using something like Glif Pools could mean the same FIL used to lock as collateral is also being used by the auction itself.
    - This removes the opportunity cost of locking up FIL here vs for other yield bearing use cases.

Things that might be tricky here:

- Flow of datacap issuing
- Ensuring the right data is onboarded
- Gas costs of triggering the deal
    - With the new proposal from anorth though how expensive would this be?
    - Possible to batch multiple auction wins